Conceptual design of an automated steel wall framing assembly using axiomatic design and integrated function model

Tamayo, E., Khan, Y. I., Qureshi, A. J. & Al-Hussein, M.,

Original citation & hyperlink:
Tamayo, E, Khan, YI, Qureshi, AJ & Al-Hussein, M 2019, 'Conceptual design of an automated steel wall framing assembly using axiomatic design and integrated function model', Construction Robotics, vol. 3, no. 1-4, pp. 83-101.
doi.org/10.1007/s41693-019-00022-8

DOI doi.org/10.1007/s41693-019-00022-8
ISSN 2509-811X
ESSN 2509-8780

Publisher: Springer

The final publication is available at Springer via http://dx.doi.org/10.1007/s41693-019-00022-8

Noname manuscript No.
(will be inserted by the editor)

Conceptual design of an automated steel wall framing assembly using axiomatic design and integrated function model

· · ·

Received: date / Accepted: date

Abstract The design of a manufacturing system involves multiple technical disciplines consisting primarily of mechanical, electrical, instrumentation and control engineers. A function modeling methodology in conceptual design ensures that the various disciplines work toward a common design intent. A systematic approach in efficiently capturing the design intent, promotes interdisciplinary communication, clarity, and early systematic determination of a functional design that fulfills customer needs. This paper proposes a model-based systems engineering approach that combines the advantages of axiomatic design, design structure matrix, and integrated function modeling.

Keywords Integrated function modeling. Modular construction. Automated assembly. Axiomatic design. Design structure matrix.

1 Introduction

Design iterations become costlier as a project progresses due to the increasing amount of effort and resources committed to obtain greater certainty about the cost of implementing the project (MacLeamy 2004). In construction projects, where significant capital outlay is incurred, changes beyond the conceptual design phase cause an exponential rise in costs and delays in project completion. Typically, these phases in engineering are: conceptual design, front-end engineering design (FEED) or basic engineering, detailed engineering and implementation. Uppal (2003) indicates the cost estimate accuracies in conceptual, FEED, and detailed engineering to be ± 50, ± 30, and ± 10 respectively, where an increase in cost certainty reflects a corresponding increase in the design effort and resources required. To avoid costly changes, construction managers proceed through a gated approval process to ensure the design requirements are fulfilled before moving to the next phase (Chao and Ishii 2005). Engineering design packages produced during each phase provide estimates of the project cost and facilitate communication in order to fulfill the design intent of the project among the various stakeholders involved (Oberlender 2014). In an article by Rogers (2018) the schematic design phase in the architecture, engineering and construction industry is described as the stage where the architect collaborates with the client to establish project design requirements and develop concepts to meet these needs. This stage is similar to the FEED stage of mechanical engineering design projects. Similarly, in designing and implementing manufacturing systems, communication of the design intent is vital to avoid costly iterations throughout the various phases of the project (Chiu 2002). This dynamic serves to reinforce the motivation, described in (Dong and Whitney 2001), for obtaining the design information early in the design process, at which time the cost of changes is low and the positive impact on the project is high.

Model-based systems engineering (MBSE) is a methodology for avoiding costly iterations that uses diagrams as the essential framework for communication across disciplines. MBSE represents the research and design process as a flowchart like an electrical diagram. It helps to understand the decision gates, development dependencies and iterative loops of the prototyping process. Such explicit process documentation communicates the design project clearly across diverse design team to

E-mail: ()
E-mail: ()
E-mail: ()

identify risk and allocate responsibility. In contrast to CAD modeling that requires more detailed digital modeling of a project, MBSE is done during the conceptual design stage and focuses more on a diagrammatic representation of a design's function.

Axiomatic design is a matrix-based MBSE that is used by the product design team to map design parameters to functional requirements. Such mapping forms the design matrix that aids the team in determining whether or not the independence axiom for satisfactory design is accomplished. A design matrix that is uncoupled or decoupled, signifying a mapping of relationships of functional requirements and design parameters that fall within the lower triangular region, is a satisfactory design. As will be shown mathematically in Section 3, a lower triangular matrix describes the region where an uncoupled or decoupled design creates a satisfactory solution.

A mapping outside this region, is a coupled or unsatisfactory design. The axiomatic design process is useful in conceptual design but it does not consider the interaction of design parameters as design structure matrix does. Design structure matrix is another matrix-based MBSE that contains a mapping of design parameters to the same design parameters, which results in matrix that is simply called design structure matrix that describes the interaction of design parameters. As in the axiomatic design process, mapping within the lower triangular region of the matrix signifies a satisfactory design. Its limitation is mainly due to its usefulness only when more details are known about the design. Thus, the design structure matrix method is limited in its usefulness at the conceptual design phase.

Integrated function modeling is a matrix-based MBSE, as well, that is structured by a collection of matrices called, use case, process flow, actor, and state views. Due to these views, the integrated function modeling method is an ideal design framework across various disciplines. However, since the actor view is formed using the interaction matrix, or design structure matrix, it suffers the same limitation as the design structure matrix design process. The proposed design methodology combines the advantages of axiomatic design, design structure matrix and integrated function modeling. Thus axiomatic design, design structure matrix and integrated function modeling complement one another and can be combined as a design methodology in the conceptual design phase. An outline of the proposed design framework is shown in Fig. 1.

Explicitly conveying the design intent to all disciplines necessitates the use of the MBSE approach (Eisenbart 2013). The proposed methodology takes advantage of the identified strengths of axiomatic design,

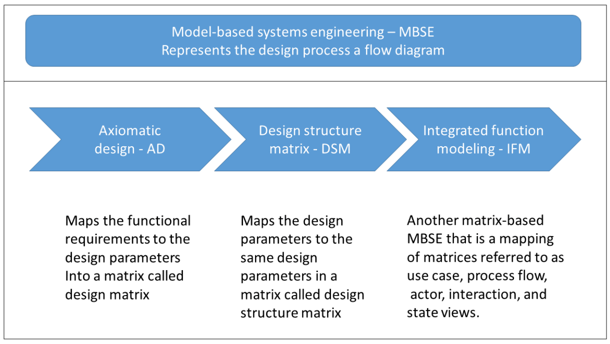

Fig. 1 Outline of the proposed integrated systems design.

design structure matrix, and integrated function modeling. Table 1 summarizes the comparison of these three design methodologies and thus describes the features of the combined MBSE methodologies for avoiding costly errors, i.e., visual, iterative, systematic, transdisciplinary, mathematical, and useful at the conceptual design phase. Details of this comparison are given in the ensuing sections.

This paper describes a three-stage MBSE design for a manufacturing system and its application to the conceptual design of an automated steel wall framing assembly. To fully describe the three-stage approach, this paper is structured as follows. Section 2 provides a state-of-the-art literature review of systeems modeling methods and the use of axiomatic design, design structure matrix, and integrated function modeling in design of manufacturing systems. Section 3 provides a discussion how this method was applied to the conceptual design of an automated steel wall framing machine. Section 4 details the mathematical formulation of this methodology. A discussion of the practical application of the proposed methodology to the conceptual design of the automated steel wall framing assembly is provided in Section 5 summarizes the lessons learned through this research and highlight areas for further development.

2 Literature review

2.1 Current state of the art in systems design in modular construction.

Bock (2015) argues that the motivation for the automation of construction is that conventional construction methodology has reached its limit. He discusses areas in construction where robotics can be deployed. For building prefabrication, he suggests automation and robotic technologies for customized components such as concrete, wood, steel, and masonry though he does not

Table 1 Comparison of Axiomatic design, design structure matrix and integrated function modeling.

Features	**Axiomatic Design**	**Design Structure Matrix**	**Integrated Function Modeling**
Useful in conceptual design phase (Dong and Whitney 2001)	√		
Useful in detailed design phase (Dong and Whitney 2001)	√	√	√
Iterative (Suh 1998, Browning 2016, Eisenbart et al. 2017)	√	√	√
Explicitly incorporates customer requirements (Suh 1997, Browning 2016, Eisenbart et al. 2017)	√		
Compact visual representation of system architecture (Abramovici and Stark 2015, Hong and Park 2009, Eisenbart et al. 2016)	√	√	√
Accommodates mathematical formulation and techniques available in the literature (Suh 1997, Browning 2016)	√	√	
Considers interaction of design parameters (Browning 2016, Eisenbart 2017)		√	√
Features integrated multidisciplinary design framework (Eisenbart 2013)			√

specify the methods for panelized wall frames. Efficiently designing automated technologies for panelized wall framing or for construction methods in general, requires function modeling methodologies to avoid costly errors.

Other MBSE solutions are unified modeling language (UML) and systems modeling language (SysML). Sudarsan et al. (2003) describe the application of a core product model to an electro-mechanical assembly using UML as a precursor to standard for the exchange of product in the lifecycle of the product. In multi-storey modular building construction, Ramaji et al. (2016) use UML to represent a product-based design methodology called product architecture model. Valdes et al. (2016) apply SysML, an extension of UML, to building construction with the objective of minimizing costly construction errors due to conflicting design specifications. Due to its inadequacy in visually representing the system architecture, as noted by Torry-Smith et al. (2011). To overcome this and other challenges associated with interactions of design parameters, this paper proposes a three-stage integrated MBSE solution involving axiomatic design, design structure matrix, and integrated function modeling. In the following sections, these design solutions are described in detail before linking them into a three-stage integrated conceptual design methodology that is visual, systematic, iterative, and transdisciplinary.

2.2 Axiomatic design

Suh (1997) describes traditional systems design as a paradigm based on know-how and trial and error, which can lead to costly errors. Suh (1998) presents the systems design theory based on axiomatic design. The axiomatic design process documents the system architecture of a design that maps the design objectives into a hierarchy of functional requirements, design parameters and process variables. The axiomatic design document incorporates fundamental principles into a process map to improve upon traditional design systems based on trial and error (Suh 1995). Axiomatic design uses the independence axiom (uncoupled or decoupled mapping) and the axiom of least information (simplicity of design) to determine a satisfactory design. Relationships between functional requirements and design parameters are marked 'X' if they exist, otherwise they are left blank, Uncoupled and decoupled design represent a diagonal and lower triangular mapping, respectively. Relationships outside the lower triangular mapping indicate a coupled design or an unsatisfactory design. An iterative design process implies redesigning the product or service such that the final mapping of functional requirements and design parameters is in the lower triangular state. This process has been used in the development of many manufacturing design processes including the design of a furniture manufacturing system (Gu et al. 2001). In this case the axiomatic design helped in the synthesis for transforming the customer needs into the mapping of a hierarchy of functional requirements, design parameters and process variables.

2.3 Design structure matrix

A design structure matrix consists of rows and columns labeled as the design parameters obtained from the axiomatic design process. The mapping procedure discussed for the axiomatic design framework applies to that of the design structure matrix method with its relationships expressed in terms of design parameters. As presented in Fig. 2, through the use of a spreadsheet, relationships and interdependencies of the design parameters are marked 'X' if they exist, otherwise they are left blank. A mapping of interdependencies that forms a lower triangular matrix provides an indication of an acceptable design, since decoupled and uncoupled interdependencies fall within the lower triangular matrix. In the case of a mapping that is not lower triangular (coupled interdendencies), a process of adding design parameters and permutation is performed until an acceptable design is reached. Design structure matrix is a design methodology developed by Steward (1981), which has been applied to the design of products, organizations and processes (Browning 2016). Its shortcoming stems from the difficulty of developing the design structure matrix at the conceptual design stage for new designs (Tang et al. 2009) due to the insufficient amount of details that are available at this stage. To overcome this shortcoming, the design structure matrix has to be derived using the design parameters obtained by forming the design matrix in the axiomatic design process.

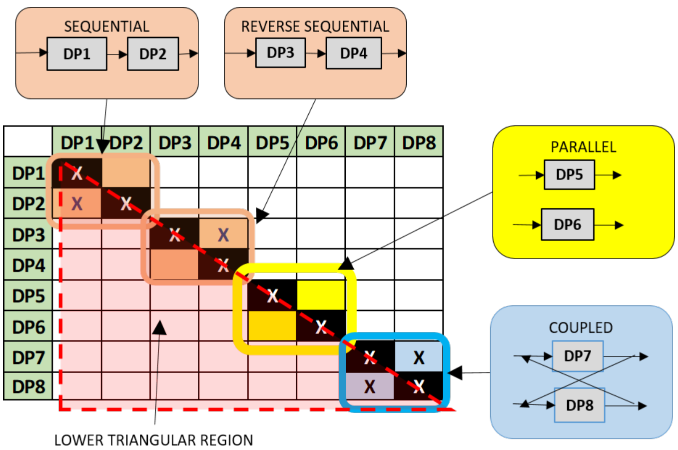

Fig. 2 Types of interactions in design structure matrix. The relationships in black are the diagonal elements of the matrix and those in the red area belong to the lower triangular region of the matrix.

Tang et al. (2009) underscore the shortcomings of axiomatic design in limiting itself to system architecture and thus its inadequacy in providing the final design solution since it does not consider the interactions among design parameters. They indicate that design structure matrix is a structure modeling method that represents the interactions among design parameters. Dong and Whitney (2001) present a technique of obtaining the design structure matrix from the design matrix derived from axiomatic design. This technique involves the building of a design matrix, appropriately selecting output variables, performing permutation to ensure that these variables are in the diagonal, and replacing the functional requirements with design parameters.

2.4 Integrated function modeling

An integrated function model consists of use case, process flow, actors, states, and interactions. This is documented by displaying the relationships and interdependencies of a set of views. This visual representation contrasts those of the axiomatic design and design structure matrix methods, which describe relationships through the use of only one view or matrix. Fig. 3 shows the views that establish the integrated function model as an effective MBSE design framework for the transdisciplinary team. These set of views are specific to only one use case. Thus, a set of views are built for each use case. It should be noted that actors are the stakeholders, hardware or software, and aspects of the environment that influence function fulfillment (Eisenbart 2015).

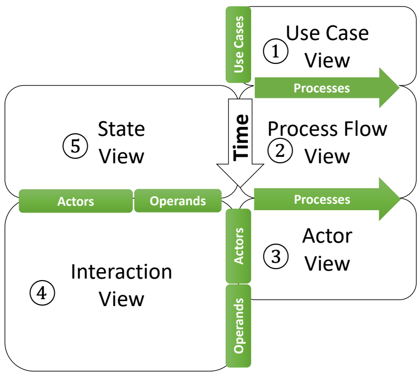

Fig. 3 Complete integrated functional modeling architecture.

The integrated function modeling process was developed after comparison of design modeling methods used by different disciplines such as mechanical and electrical engineering, mechatronics, software design, and building design (Eisenbart et al. 2011). Since every discipline's design methodology differed in structure and complexity, it would be difficult to ensure consistency and common understanding of the design intent across disciplines. By analyzing the function modeling approaches of different disciplines, Eisenbart et al. (2013) identify an integrated function modeling framework consisting of states, effects, transformation processes, interaction processes, use case, technical system allocation and stakeholder allocation.

Eisenbart et al. (2014) adopt design structure matrices in presenting the concept of integrated function modeling and use the example of a coffee vending machine in order to describe the approach. They identified three use cases for the coffee vending machine, namely: (1) preparing cappuccino, (2) preparing hot water and (3) automated cleaning. For the second use case, they describe its process flow when coffee is ordered, water is heated and coffee beans are ground simultaneously before mixing the howt water and ground coffee together. After mixing, the cup is filled with the mixture and the waste is disposed of. By presenting this process flow in the integrated function modeling process, each discipline can design the shared understanding of the design intent.

Their elaborated integrated function modeling framework includes use case, transformation processes, interaction processes, effects, states, technical subsystems, stakeholder and environment. They have outlined further enhancement of the integrated function modeling approach using a software tool that will automate the design modeling.

3 Three-stage design methodology

An integrated axiomatic design, design structure matrix, and integrated function modeling systems engineering solution provides an effective representation of the system architecture of a manufacturing system, such as the steel wall framing assembly, due to the following advantages: (1) it provides a compact visual representation of the system architecture (this feature is important in ensuring that updates to the documentation consistently accommodate constantly changing models at the conceptual design phase (Abramovici and Smart 2013, Hong and Park 2009, Eisenbart et al. 2016); (2) promotes creativity in the application of fundamental principles and mapping of the design objectives into functional requirements, design parameters, and process variables (Suh 1995); (3) clearly communicates interactions among design parameters, transformation processes, use cases, and states (Eisenbart et al. 2015); and (4) provides mathematical support to the resulting integrated function modeling framework due to the integration of axiomatic design and design structure matrix, which have mathematical bases (Suh 1995, Browning 2016).

The proposed systems engineering design of manufacturing systems is best applied in the conceptual design phase, i.e., the phase where models are most likely to change and where changes are least costly to make. This paper focuses on the application of the integrated axiomatic design and integrated function modeling systems design approach to the conceptual design of manufacturing systems, particularly the case of an automated steel wall framing assembly.

The model-based systems engineering approach to the design of a manufacturing system is best applied in the conceptual phase where designs are most likely to change without incurring significant cost. This paper focuses on the application of the integrated axiomatic design, design structure matrix and integrated function model during the conceptual design of manufacturing systems, in this case a machine for automated assembly of steel framed walls. this process documents the design drivers of the project before a design is sketched on paper or a digital 3D model is built. This framework ensures that the design sketches and 3D CAD models are efficiently coordinated during the design development. The model-based systems engineering utilizes flow charts and process diagrams while the detailed design process implements CAD tools such as CATIA or SOLIDWORKS for the development of architectural or mechanical systems.

3.1 Stage 1

A summary of the steps in the axiomatic design stage, also depicted in Fig. 4, is provided below:

1. Identify the customer needs
2. Map the customer attributes onto the functional domain of functional requirements
3. Map the functional domain of functional requirements onto the physical domain of design parameters
4. Check if the independence axiom is satisfied
5. Revise the design
6. Choose the best design, or least information content, if there are multiple designs

Axiomatic design explicitly documents customer needs in terms of assessing different product use cases to determine functional requirements and considers how to achieve them through the features of the design object. These features are considered design parameters and can be changed in order to evaluate which design option best fulfills the functional requirement of the customer

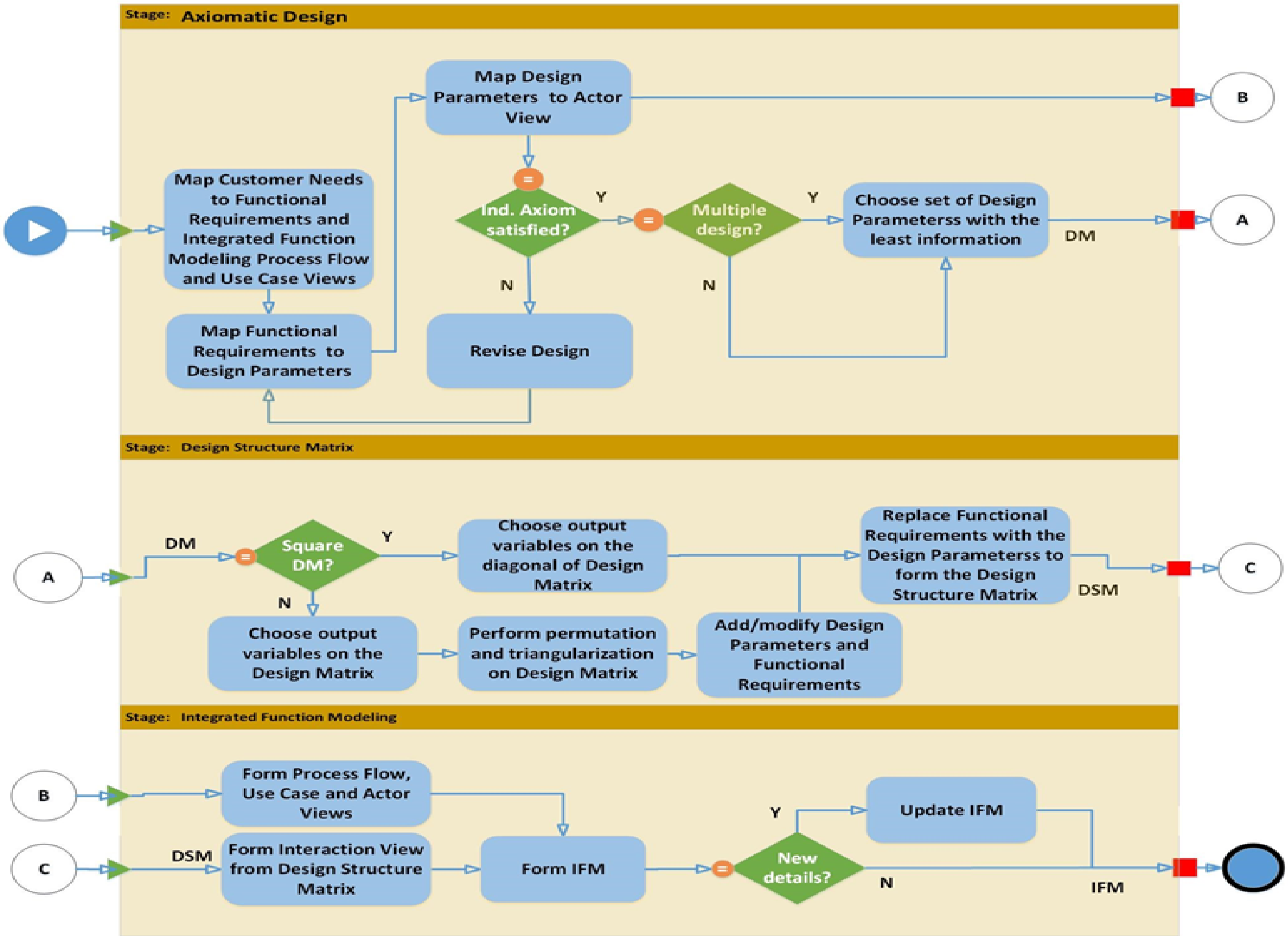

Fig. 4 The three-stage approach.

needs. The relationship between these can be described mathematically to quantify the decision-making process. A description of this mathematical assessment is presented as follows.

Following the work of Suh (1998) on mapping of the functional domain onto the physical domain, the design matrix establishes the relationship between the functional requirements and design parameters in binary notation expressed as

$$\{FR\} = [DM]\{DP\} \tag{1}$$

$$DM_{ij} = \begin{cases} X, if\, an\, element\, or\, effect\, exists \\ 0, otherwise \end{cases} \tag{2}$$

where $i, j = 1 \cdots n$.

To examine the impact of adjusting a design parameter to an functional requirement, considering all other design parameters as constant, design matrix is expressed in terms of sensitivities $\frac{\partial FR}{\partial DP}$ and incremental functional requirements and design parameters:

$$\{\Delta FR\} = [DM]\{\Delta DP\} \tag{3}$$

$$DM_{ij} = \begin{cases} X, & if \frac{\partial FR}{\partial DP} \neq 0 \\ 0, & otherwise \end{cases} \tag{4}$$

Depending on the system being considered in design, the design matrix in Equation (4) takes on binary or transfer function entries. In the design of feedback controllers, for example, functional requirements and design parameters are analogous to controlled variables and manipulated variables, respectively. Controlled variables refer to the signals from the sensors that are fedback to the controller, while manipulated variables are the controller outputs that drive a system to achieve the desired objectives. Given this association, it is interesting to note that the design of controllers is axiomatic, since this methodology also uses a design matrix in binary form during the brainstorming sessions of the conceptual design phase. If the dynamic model of the system to be controlled is known, however, the design matrix consists of transfer functions relating the manipulated variabless with the controlled variables. This design matrix, expressed in Laplace transforms, clearly communicates the behavior of the process to be controlled to the multidisciplinary team comprising the different stakeholders of the control design project. Once the dynamic design matrix is established, the multivariable controller is also established, because the controller is essentially the inverse of the open-loop design matrix. A system with a diagonal design matrix is easiest to control compared to those with coupled or decoupled design matrices since the multivariable controller mainly consists of independent single-input-single-output (SISO) controllers. A SISO system provides an ideal situation that allows the simplest implementation of a multi-loop controller. Similarly, a SISO system in axiomatic design (Farid and Suh 2016) is an uncoupled design with a diagonal design matrix, which is concisely expressed as

$$DM_{ij} = \begin{cases} X, & if\, i = j \\ 0, & otherwise \end{cases} \tag{5}$$

An uncoupled design satisfies the independence axiom (Suh 1998). A lower triangular design matrix is a decoupled design that also satisfies the independence axiom as expressed below.

$$DM_{ij} = \begin{cases} X, & if\, i < j \\ 0, & otherwise \end{cases} \tag{6}$$

$$\begin{aligned} \Delta FR_1 &= f(\frac{\partial FR_1}{\partial DP_1}\Delta DP_1) \\ \Delta FR_2 &= f(\frac{\partial FR_1}{\partial DP_1}\Delta DP_1, \frac{\partial FR_2}{\partial DP_2}\Delta DP_2) \\ &\vdots \\ \Delta FR_i &= f(\frac{\partial FR_1}{\partial DP_1}\Delta DP_1, \cdots, \frac{\partial FR_i}{\partial DP_i}\Delta DP_i) \end{aligned} \tag{7}$$

ΔFR_i in Equation (7) is satisfied since it uses previously determined DPs, $DP_1 \cdots DP_{i-1}$, and its corresponding DP, DP_i.

A coupled design, on the other hand, does not satisfy the independence axiom. Below is an example of a coupled design with full matrix design matrix.

$$DM_{ij} = X, for\, all\, values\, of\, i\, and\, j \tag{8}$$

thus

$$\begin{Bmatrix} \Delta FR_1 \\ \vdots \\ \Delta FR_n \end{Bmatrix} = \begin{bmatrix} \frac{\partial FR_1}{\partial DP_1} & \cdots & \frac{\partial FR_1}{\partial DP_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial FR_n}{\partial DP_1} & \cdots & \frac{\partial FR_n}{\partial DP_n} \end{bmatrix} \begin{Bmatrix} \Delta DP_1 \\ \vdots \\ \Delta DP_n \end{Bmatrix} \tag{9}$$

$$\begin{aligned} \Delta FR_1 &= f(\frac{\partial FR_1}{\partial DP_1}\Delta DP_1, \cdots, \frac{\partial FR_1}{\partial DP_n}\Delta DP_n) \\ &\vdots \\ \Delta FR_i &= f(\frac{\partial FR_i}{\partial DP_1}\Delta DP_1, \cdots, \frac{\partial FR_i}{\partial DP_n}\Delta DP_n) \end{aligned} \tag{10}$$

Equation (10) reveals that a functional requirement of a coupled design is difficult to control since more than one design parameter has influence over it. In this case, a direction must be provided to resolve the coupling issues (Do and Park 2001). Coupled designs necessitate a deeper analysis of the design parameters in creating new design parameters or choosing the best design parameters (Farid and Suh 2016).

If there are multiple designs and the independence axiom is satisfied for each design, such as in the choice of DC motor, an engine, or a combination of DC motor and engine (hybrid) for prime movers, the best design is considered to be the one with the least information content (Do and Park 2001), or

$$I_{min} = min \begin{Bmatrix} n \\ \Sigma I_i \\ i = I \end{Bmatrix} \tag{11}$$

where

$$\begin{aligned} I_i &= log_2 \frac{1}{p} \\ &= log_2 \left(\frac{system\, range}{Common\, range} \right) \end{aligned} \tag{12}$$

In Equation (12), p is the probability of satisfying the functional requirement FR_i. Equation (12) reflects the following three points: (1) simplicity in design is associated with the least information satisfying the functional requirements; (2) a simple design ensures a high probability of success in achieving the functional requirements, since, if p is at maximum, or equal to 1, then the information content I is 0; and (3) a simple design is fulfilled if functional requirements are consistently satisfied without bias.

Fig. 4 illustrates the significance of the axiomatic design stage due to its iterative procedure. It has been noted that the best phase to detect design errors and make changes is the early design phase such as FEED and schematic, where the cost impact of changes is still low. In the axiomatic design stage the design goals are fully communicated and design loopholes are identified and corrected through the iterative process. However, a process for considering the interaction among the design parameters in order to finalize the design still does not exist in axiomatic design. This design process is discussed in the next section.

3.2 Stage 2

At the the previous axiomatic design stage, the process flow, use case, and actor views of the integrated function modeling architecture are initiated. In the second stage or design structure matrix stage, a method for building the interaction view, depicted in Fig. 3, in integrated function model by deriving the design structure matrix from the design matrix obtained in stage 1 is described. It should be noted that that the final design parameters derived in this stage become the actors in the interaction view of Fig. 3.

Information about the interactions among the design parameters is represented in a matrix when decomposed is referred to as design structure matrix method. Sequential, parallel, and coupled interactions are the different types of interactions in design structure matrix, described in Fig. 2. Sequential interaction refers to an attainable design parameter due to the availability of previous information displayed as marked relationships below the diagonal. Reverse sequential depicts an unattainable design parameter due to the absence of of previous information shown as marked relationships above the diagonal. Parallel interaction pertains to independent design parameters. Coupled interaction is either sequential or reverse. Among these types of interactions, the reverse sequential and coupled interactions are undesirable due to the assumptions made by the preceding design parameters to carry out their tasks. Decomposition of interaction matrices into lower triangular matrices in design structure matrix can be achieved by minimizing the coupling of design parameters through clustering, tearing, or triangularization (Guenov and Barker 2005).

Coupled interactions such as that presented in Fig. 2 may be unavoidable. For example, consider a position control mechanism in a closed-loop system consisting of a DC motor equipped with an encoder. The target position is provided as a setpoint, and the controller drives the motor to this setpoint based on the feedback information provided by the encoder. Relating this example to Fig. 2, $G(s)$ and $H(s)$ are the motor assembly and control system transfer functions, respectively. Coupling between $G(s)$ and $H(s)$ signifies a feedback control signal, $Y(s)$, through the encoder that corrects the position of G(s) until the setpoint, $R(s)$, is achieved. This closed-loop system is depicted in Fig. 5. Evidently, a lesson can be drawn from this automatic feedback system: that, for complex systems, couplings are clustered into smaller independent modules to enable faster corrections to the assumptions made at the outset. A performance indication of the speed at which the feedback signal approaches the target is described by the closed-loop transfer function in Equation (13). If $K(s)$ is the controller transfer function for this control system, $H(s)$, then the closed-loop transfer function of this feedback system is:

$$\frac{Y(s)}{R(s)} = \frac{K(s)G(s)}{1 + K(s)G(s)} \tag{13}$$

The axiomatic design phase having been completed, the design matrix is then finalized in the design structure matrix stage. This proces of passing the design matrix to the design structure matrix stage is depicted by the connector A in Fig. 4. The dominant design parameter for each functional requirement is chosen as the output variable of each row of the design matrix. For square design matrix, the output variables are simply the diagonal design parameters; (refer to the proof of this assertion in Dong and Whitney (2001). If design matrix is not square, however, the rows are permuted and design parameters and functional requirements are added or modified while placing the output variables on the diagonal. This permutation is combined with triangularization and the functional requirements are replaced with the design parameters of the columns to obtain the design structure matrix (Guenov and Barker 2005). As in axiomatic design, design structure matrix involves any of the forms discussed for design matrix, such as the following equations for a lower triangular matrix:

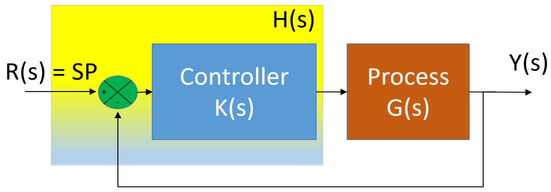

Fig. 5 Coupled interaction for controller, K(s) and process, G(s) in a feedback control loop.

$$\{DP\} = [DSM]\{DP\} \tag{14}$$

where

$$DSM_{ij} = \begin{cases} 0, & if\, i < j \\ X, & otherwise \end{cases} \tag{15}$$

For an $n \times n$ DSM,

$$\begin{Bmatrix} DP_1 \\ \vdots \\ DP_n \end{Bmatrix} = \begin{bmatrix} X & & \\ \vdots & \ddots & \\ X & X & X \end{bmatrix} \begin{Bmatrix} \Delta DP_1 \\ \vdots \\ \Delta DP_n \end{Bmatrix} \tag{16}$$

3.3 Stage 3

Integrated function modeling is constructed from the resulting interaction matrix of previous processes. Integrated function modeling compactly displays its information in matrices as detailed in Fig. 3.

Thus, integrated function modeling represents a complete picture of the system design for a cross-disciplinary group of technical and nontechnical stakeholders. Based on a survey with designers from various companies, Eisenbart et al. (2015) deem integrated function modeling to be useful in: (1) modeling and analyzing actor dependencies, (2) analyzing environmental impact, (3) analyzing function time dependencies, and (4) analyzing model consistency and completeness. The integrated function modeling is a working framework that captures the interdisciplinary perspectives and facilitates communication of the design goals. Fig. 3 illustrates this framework.

Integrated function modeling portrays a comprehensive but compact picture of the design since it encompasses the results obtained from axiomatic design and design structure matrix as well as any other information involved in the design. The procedure for forming the integrated function modeling is presented in Fig. 4 and summarized in Table 2.

Table 2 Steps as outlined in Fig. 3 for constructing the integrated function modeling, (B. Eisenbart et al. 2015).

Steps	Integrated Function Modeling Display View	Description
①	Use case view	Lists the applications of the design and is built at the axiomatic design stage.
②	Process flow view	Describes the main processes for a specific use case and is built at the axiomatic design stage. Mathematically, the process flow view can be expressed in first order logic.
③	Actor view	Shows the assignment of design parameters that are used to satisfy the processes. This view is initially built at the axiomatic design stage using the first-level design parameters formed when mapping the functional, requirements to the design parameters.
④	Interaction view	Consists of operands and actors, which are mainly the final design parameters obtained from the design structure matrix stage. Operands are energy, material and signal inputs.
⑤	State view	Depicts the change in state or transformation caused,by actors (design parameters) and operands (inputs) as the system goes through a series of processes to realize a finished product. This view completes the construction of the integrated function modeling.

3.4 A Simple illustration

Describing the initial design of the automated steel wall framing machine provides a simple illustration of the basic steps of the integrated design approach depicted in Fig. 4. For illustrative purposes, this example only considers the high-level functional requirements (FRs) of the steel wall framing assembly depicted in Fig. 6. A detailed illustration of the integrated design approach of the same steel wall frame machine, which includes addressing a potential coupling concern, will be provided in the next section.

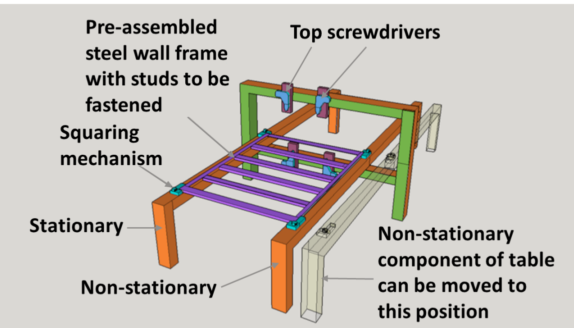

Fig. 6 Isometric view of steel wall framing assembly.

lists the high-level FRs that result from the product design team's effort in determining the customer requirements (CRs), which are also referred to as customer needs. At the axiomatic design (AD) stage in Fig. 4, the design team forms the design matrix (DM) by mapping the FRs to the following design parameters (DPs): automated wall framing assembly (DP0), machine for three steel frame types (DP1), right table movement (DP2), manual assembly (DP3), squaring system (DP4) and dragging system (DP5).

As previously mentioned, the process, use case, and actor views are simultaneously constructed for the integrated function modeling (IFM) with the DM. Referring to Table 3, mapping of the customer needs for the steel wall framing assembly to the high-level FRs immediately results in the process and use case views of the IFM.

Table 3 Mapping of high-level functional requirements, processes and use cases.

Customer Needs	Mapping		Use Case	
	Axiomatic Design	Integrated Function Modeling	No.	Label
Make automated steel wall frame machine	FR0: Provide automated steel wall framing assembly			
Provide machine for 3 wall frame types	FR1: Provide machine for three steel types	P1: Input frame information	1	Make frame with studs only
			2	Make frame with studs and window
			3	Make frame with studs and door
Provide machine for different widths	FR2: Provide at least 2DOF in the machine	P2: Table frame positioning		
Manual assembly	FR3: Provide means for manual access and assembly	P3: Pre-assembly		
Square wall frame	FR4: Provide squaring system	P4: Squaring		
Self-drilling screw fastening	FR5: Provide self-screw fastening system	P5: Screw fastening		

A high-level DM for the simple design example of the automated steel wall frame machine depicted in Fig. 6 is formed using Equations (1) and (2) as follows:

$$\begin{Bmatrix} FR0 \\ FR1 \\ FR2 \\ FR3 \\ FR4 \\ FR5 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 & 0 & 0 \\ 0 & 0 & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & 0 & X \end{bmatrix} \begin{Bmatrix} DP0 \\ DP1 \\ DP2 \\ DP3 \\ DP4 \\ DP5 \end{Bmatrix} \tag{17}$$

Equation (17) implies that the independence axiom of Fig. 4 is satisfied and that the design team can then proceed in forming the high-level DSM by replacing the FRs with their corresponding DPs. In addition to the operator, these DPs become the actors of the interaction view of the IFM in Fig. 7. Representing another aspect of the interaction view, are the operands that interact with the actors of the steel wall framing machine are the electricity, control system commands, steel channel, 80/20 profile, and screws. How the actors influence or are influenced by the processes are marked as 'X' or 'O', respectively, in the actor view. The state view shows the change of states of the actors and operands when an actor or a group of actors executes a process.

4 Application to automated steel wall framing assembly

Application of the three-stage approach to the conceptual design of a steel wall framing assembly begins with compiling a list of customer needs:

1. Make automated steel wall frame assembly
2. Manually pre-assemble wall frame prior to the start of the automated process
3. Fasten pre-assembled wall frame using self-drilling screws
4. Provide a machine that can accommodate at least three wall frame types: studs only, stud with window, and studs with door
5. Provide a wall frame that is properly squared

The following sections provide the low-level details of building the IFM with the aid of the three-stage design methodology.

4.1 Formulating the design matrix

Breaking down the FRs further indicates the need to conceive a design solution with the corresponding DPs. Mapping of FRs into DPs of the potential design solution presented in Fig. 6 is provided in Table 4. The functional design of the steel wall framing assembly presented in Fig. 6 includes the following components: (*i*) two tables (A and B) with one side moveable to accommodate various wall frame widths; (*ii*) top and bottom gantries to hold the power screwdrivers; and (*iii*) squaring mechanisms.

Two sets of power screwdrivers drive screws into the top and the bottom of the pre-assembled wall frame. Positions of these screwdrivers on the y-axis are based on the input recipe of wall frame type, width, and use case. These two sets of screwdrivers move along the z-axis to drive self-drilling screws into the panel. Positions vary depending on the type of panel to be fabricated.

The feedback control system is a coupled system that is a necessary and acceptable interaction. However, another type of coupling is identified in the self-drilling screw fastening process that violates the desired property of a lower triangular DM. This coupling can be initially shown considering the gantry setup for one screw fastening operation represented by an arm consisting of two prismatic joints ① and ②, and one revolute joint ③, referred to as a PPR arm, shown in Fig. 8. Prismatic joint ① represents the positioning of the screw driver along the y-axis while prismatic joint ② represents the positioning of the screw driver along the z-axis. Screw fastening consists of two simultaneous movements of joints ② and ③. Joint ③ describes the revolute action of the end effector, which is the tip of the screw driver that drives the screw into the frame. Simultaneous action of two joints signify a coupled relationship that is analyzed using the simple engineering principles outlined in Table 4.

Re-ordering and triangularization techniques are used to make the DM lower triangular as much as possible. Table 4 indicates the coupling derived from the equations governing the torque required to drive the self-drilling screw into the frame and its corresponding thrust, as well as the torque required to drive the screwdriver along the z-axis.

New set of FRs and DPs are created and the process of re-ordering, re-numbering, and triangularization produces a lower triangular DM. Although related to the z-positioning system, the added FRs and DPs signify a software solution that produces the necessary torque to provide the required screwdriver thrust. Fig. 9 presents the DM resulting from the steps just described.

4.2 Completing the integrated function modeling

Use case, process, and actor views are developed in formulating the DM, which is in turn used to build the DSM. Following the DSM stage procedure provided in Fig. 4, the output variables are simply the diagonal elements, since the DM is square. Thus, the DSM is the DM represented in Fig. 9 with the FRs replaced by the DPs. Once the DSM is built, the interaction view is built, leaving the state view as the only display remaining to be constructed. To complete the IFM, the Process Flow, Use Case, and Actor Views obtained from the AD stage are updated using the output of the DSM stage as illustrated in Fig. 4.

Initial and final states of the state view are the basis of the transformation logic of each of the steel wall framing assembly processes. At a high level, the state view communicates insight on how to control the processes to satisfy the design goals. This view provides a programming framework for the software or control aspect of the design.

High-level results of the completed three-stage design method for the automated steel wall framing assembly can be observed in the IFM framework presented in Fig. 7. For readability, sections of the detailed

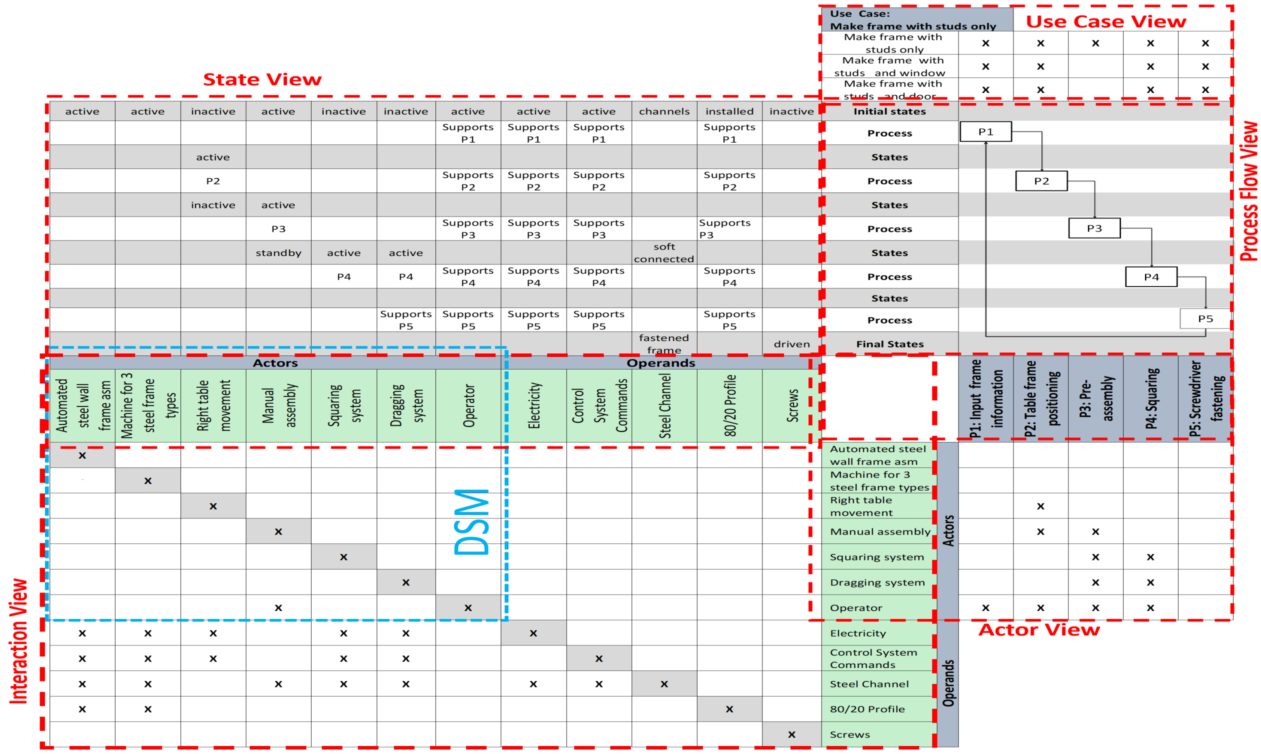

Fig. 7 Integrated Function Model incorporating the Design Structure Matrix of the high-level design example of the automated steel wall framing machine. Processes are: Input frame information (P1), Table frame positioning (P2), Pre-assembly (P3), Squaring (P4), and Screwdriver fastening (P5).

Table 4 Coupling of the functional requirements and design parameters of the fastening system.

Functional Requirements	Design Parameter	Equation	Comments
Provide z-axis screwdriver positioning system	z-axis screwdriver positioning system	$M_z = f(T, p)$ (18)	Torque, M_z, of the positioning system is a function of the screw fastening thrust, T, and the geometrical parameters p, of the drive mechanism.
Provide fastening system	Fastening system	$T = 2K_d F_f F_T BW + K_d D^2 JW$ (19) $M = 2K_d F_f F_M AW$ (20)	T = screw fastening thrust and M = screwdriver torque. Equations (19) and (20) are taken from Oberg et al. (2016)

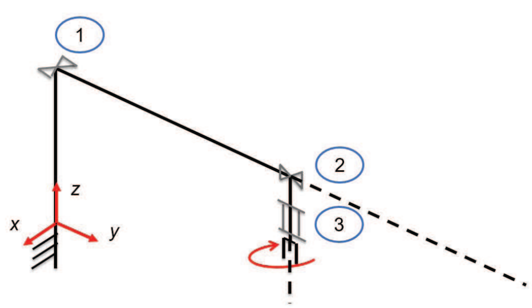

Fig. 8 PPR arm representation of a gantry setup for a single screw driver with prismatic joints ① and ② and revolute joint ③.

IFM are presented in Appendix B, namely (i) Fig. 10 Use case and process flow views, (ii) Fig. 11 Actor view, (iii) Fig. 12 Interaction view, and (iv) Fig. 13 State view.

As indicated in Table 5, the process flow view can be mathematically expressed in first order logic (Russel and Norvig 2016). This formulation will be useful in implementing the processes on any programmable controller platform. For the first order logic, the variables are defined as follows:

$F = \{f_i \mid i \in [1, n]\}$; a set of n frames to be produced.

		DP0 Automated steel wall frame assembly	DP1 Machine for 3 steel types	DP1.1 User Interface	DP1.2 Control Interface	DP1.3 Recipe Data	DP2 Right table movement	DP2.1 Right table mechanism	DP2.1.1 Right table mechanism control	DP3 Manual assembly	DP4 Squaring system	DP4.1 Holding mechanism	DP4.1.1 Holding control	DP4.2 Right angle positioning system	DP4.2.1 Right angle positioning control	DP5 Dragging system	DP5.1 Dragging system control	DP6 Y-axis screw driver positioning system	DP6.1 Y-axis screw driver positioning control	DP7 Z-axis screw driver positioning system	DP7.1 Z-axis screw driver positioning control	DP8 Fastening system	DP8.1 Fastening system control	DP9 Z-axis thrust postion	DP9.1 Z-axis thrust postion control
FR0	Provide automated steel wall frame assembly	X																							
FR1	Provide machine for three steel types		X																						
FR1.1	Provide recipe interface			X																					
FR1.2	Provide recipe access			X	X																				
FR1.3	Provide recipe			X	X	X																			
FR2	Provide right table movement						X																		
FR2.1	Provide right table mechanism							X	X																
FR2.1	Provide right table mechanism control							X	X																
FR3	Provide manual assembly									X															
FR4	Provide squaring system										X														
FR4.1	Provide holding mechanism											X	X												
FR4.1	Provide holding control											X	X												
FR4.2	Provide right angle positioning system													X	X										
FR4.2	Provide right angle positioning control													X	X										
FR5	Provide dragging system															X	X								
FR5.1	Provide dragging system control															X	X								
FR6	Provide y-axis screw driver positioning system																	X	X						
FR6.1	Provide y-axis screw driver positioning control																	X	X						
FR7	Provide z-axis screw driver positioning system																			X	X				
FR7.1	Provide z-axis screw driver positioning control																			X	X				
FR8	Provide fastening system																					X	X		
FR8.1	Provide fastening system control																					X	X		
FR9	Provide z-axis thrust postion																					X		X	X
FR9.1	Provide z-axis thrust postion control																							X	X

Coupling due to feedback control system

Coupling shifted by adding FR9 and FR9.1 with their corresponding DPs

Fig. 9 Final design matrix of the automated steel wall framing assembly.

$X = \{x_j \mid j \in [1, m]\}$; a set of m x coordinates, defining the position where the screw will be applied to the frame along the x-axis.
$Y = \{y_j \mid j \in [1, m]\}$; a set of m y coordinates, defining the position where the screw will be applied to the frame along the y-axis.
Ls = length of screw; this is the vertical distance the screw will be traversing through the steel channel.
Wd = width of frame to be produced. This is contained in the use case (frame information).
Zsd = Travel of screw driver in the z direction. This is contained in the use case.

Functions and first order logic describing the processes are listed in Table 5.

In forming the actor view of Fig. 11, inputs or operands are added such as those required for realizing the automated steel wall framing assembly. These operands, which complement the previously identified DPs, consist of an operator who oversees the manufacturing process, electricity to power the system, a control system which constitutes the SCADA/sensor level, a steel channel for making the wall frames, the 80/20 profile (from 80/20 Inc.) from which the assembly is built, and the self-drilling screws for fastening the wall frames. An effect is marked by an 'X' if an actor or an operand directly affects a process, and by an 'O' if a process affects an actor or operand. Fig. 12 presents the DSM

Table 5 Functions describing the automated steel wall framing machine.

Function	Description
$InputControllerUseCase(F \times Ws \times Zsd \times X \times T)$	Controller stores the use case information for use in the subsequent processes of producing a wall frame.
$PositionControllerTable(F \times Wd)$	Controller positions the table to the width Wd.
$AssembleOperator(F)$	Operator manually assembles the frame.
$DragController(F \times X)$	Controller drags frame to x coordinate.
$PositionControllerScrewDriverY(F \times Y)$	Controller positions the screw driver in y coordinate.
$PositionControllerScrewDriverZ(F \times Zsd)$	Controller positions the screw driver to vertical distance Zsd.
$DriveControllerScrew(F \times Ls)$	Controller drives screw through its length Ls.
Fist Order Logic: $D = D \cup \{F, X, Y\}$ $\forall f \in F : (\forall x \in X; \forall y \in Y.InputControllerUseCase(f, Zsd, Wd, x, y)$ $PositionControllerTable(f, Wd) \wedge AssembleOperator(f) \wedge DragController(f, x) \wedge$ $DriveControllerScrew(f, Ls)$	

with additional information on how the operands affect the system.

5 Discussion

Adopting an integrated design methodology that facilitates a collaborative thinking process across a multidisciplinary team is the motivation behind the use of the IFM framework in conceptual design (Eisenbart et al. 2011). A simple illustration of the initial design of an automated steel wall framing assembly describes how the IFM evolves from mapping the customer needs at the AD stage. Defining customer needs and determining FRs constitutes the first and crucial step in conceptual design (Suh 1997). At this initial phase, the product design team works on a solution that is easily transcribed into matrices such as the DM in AD and the process and use case views in IFM. As the design team presents more details, the DM matures into a visual representation of a desired or undesired function as illustrated in Fig. 9. In this figure, an undesired coupling has been identified and analyzed, through the use of basic engineering principles listed in Table 4, to arrive at additional FRs and DPs in resolving the conflict. This solution demonstrates the visual and iterative advantages of the matrix-based integrated design methodology proposed in this paper. Moreover the thinking process encourages creativity within the team in arriving at a solution. The thinking process is carried out through brainstorming sessions that generate many ideas (Foley and Hardardóttir). As noted in (Eisenbart et al. 2015), the application of the matrix-based integrated design methodology to the automated steel wall framing machine requires less modeling effort compared to a diagram-based framework such as SYSML. Social interaction is crucial throughout the entire conceptual design stage, but it is especially critical at the initial phase when the product design team is defining the CRs with internal and external stakeholders. Mapping of CRs to FRs shown in Table 3 and the determination of the optimal number of FRs can be time consuming if it is not done systematically. An automated implementation of the proposed methodology to the conceptual design of the steel wall framing assembly could have facilitated the thinking process more efficiently.

6 Conclusion and future work

This paper proposes an integrated design approach to the conceptual design of an automated modular construction manufacturing system and applies it to a prototype of an automated steel wall framing assembly under development at the University of Alberta, Canada. This systematic approach, consisting of AD and DSM, provides a mathematical basis and iterative design methodology for the IFM framework originally introduced by Eisenbart et al. (2012). Although the ultimate design decision is the responsibility of the design team, the methodology described in this paper facilitates decision making based on CRs. Since the three-stage approach is iterative, it is a favorable method in the conceptual design phase where the design iterations have minimal cost impact. It has been demonstrated that early detection of (and solutions to) design complexity arise from the application of basic engineering principles, even if the design parameters and functional requirements are expressed at a high level in the axiomatic design stage of the method. Due to its simplicity, the proposed matrix-based integrated design approach, which is essentially an IFM framework, is faster to develop and requires less training and modeling efforts compared to the diagram-based UML or SYSML (Eisenbart 2015). Microsoft Excel has been used to implement the integrated design of the automated steel wall framing assembly. However, future projects can benefit from the application of the proposed design methodology with the aid of other software tools. Moreover, the mapping of CRs to FRs dand the determination of the optimal number of FRs can be efficiently automated in the future using state-of-the-art techniques to achieve quality function deployment during brainstorming sessions. [1]As architects and engineers extend their interests into design to manufacturing processes they become more involved in the manufacturing of automated machine processes that can fulfill customer needs. To enable this process our research seeks to educate users about the requirements of functional modeling for development of automated manufacturing equipment. Fig. 14 in Appendix B shows the realization of the steel wall framing machine outlined in Fig. 6, which the integrated function modeling has been applied to.

References

1. Abramovici, M. and Stark, R. (2013): Smart Product Engineering. In Proceedings of the 23rd CIRP Design Conference, Bochum, Germany.
2. Adams, K.M. (2015): Nonfunctional Requirements in Systems Analysis and Design (Vol. 28). Cham, Switzerland: Springer.
3. Bock, T. (2015):The Future of Construction Automation: Technological Disruption and the Upcoming Ubiquity of Robotics. Automation in Construction 59, 113-121.

[1] quality function deployment is widely adopted in engineering practice and technical activities (Eisenbart et al. 2017).

4. Browning, T.R. (2016): Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. IEEE Transactions on Engineering Management 63(1), 27-52.
5. Chao, L.P. and Ishii, K. (2005): Design Process Error-Proofing: Benchmarking Gate and Phased Review Life-Cycle Models. In ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. 301-310). American Society of Mechanical Engineers.
6. Chiu, M.L. (2002): An organizational view of design communication in design collaboration. Design studies, 23(2), pp.187-210.
7. Do, S.-H. and Park G.J. (2001): Application of Design Axioms for Glass Bulb Design and Software Development for Design Automation. Journal of Mechanical Design. 123(3), 322-329.
8. Dong, Q., and Whitney, D.E. (2001): Designing a Requirement Driven Product Development Process. Proc. 13th International Conference on Design Theory and Methodology, 1-11.
9. Eisenbart, B., Gericke, K. and Blessing, L. (2011): A Framework for Comparing Design Modelling Approaches Across Disciplines. Proceedings of the 18th International Conference on Engineering Design ICED'11, Vol. 2, 344-355.
10. Eisenbart, B., Blessing, L. and Gericke, K. (2012): Functional modelling perspectives across disciplines: a literature review. In DS 70: Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, 847-858.
11. Eisenbart, B., and Gericke, K. and Blessing, L.(2013):. An Analysis of Functional Modeling Approaches across Disciplines. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM), 27(3), 281-289.
12. 89.
Eisenbart, B.,Gericke, K., and Blessing, L. (2014): Application of the IFM Framework for Modelling and Analysing System Functionality. In DS 77: Proceedings of the DESIGN 2014 13th International Design Conference: 153-162.
13. Eisenbart, B.,Gericke, K., and Blessing, L. (2015): DSM for Modeling and Analyzing Functionality: Views of Practitioners. Modeling and Managing Complex Systems. Proceedings of the 17th International DSM Conference (November).
14. Eisenbart, B., C. Mandel, K. Gericke, and L. Blessing. (2015): Integrated Function Modelling: Comparing the IFM Framework with SYSML. Proceedings of the International Conference on Engineering Design, ICED 5 (DS 80-05): 1-12.
15. Eisenbart, B., Khan, Y.I. and Qureshi, A.J. (2016): INTEGRATED PRODUCT MODELLING THROUGH IFM-CPM/PDD. In DS 84: Proceedings of the DESIGN 2016 14th International Design Conference (pp. 183-192).
16. Eisenbart, B., Gericke, K., Blessing, L., and Mcaloone, T. (2017): A DSM-Based Framework for Integrated Function Modelling: Concept , Application and Evaluation. Research in Engineering Design 28(1): 25-51.
17. Foley, J.T. and Hardardóttir, S. (2016): Creative Axiomatic Design. Procedia CIRP, 50, 240-245.
18. Farid, A., and Suh, N. (2016): Axiomatic Design in Large Systems.:complex products, buildings and manufacturing systems.New York, Springer.
19. Gu, P., Rao, H., and Tseng, M. (2001): Systematic Design of Manufacturing Systems Based on Axiomatic Design Approach. CIRP Annals - Manufacturing Technology 50(1): 299-304.
20. Guenov, M.D., and Barker, S.G. (2005): Application of Axiomatic Design and Design Structure Matrix to the Decomposition of Engineering Systems. Syst. Eng. 8(1), 29-40.
21. Hong, E.P. and Park, G.J. (2009): Decomposition process of engineering systems using axiomatic design and design structure matrix. In The Fifth International Conference on Axiomatic Design (pp. 25-27).
22. MacLeamy, P. (2004): MacLeamy curve, in Collaboration, Integrated Information, and the Project Lifecycle in Building Design and Construction and Operation (WP-1202). CURT, August, 2004.
23. Oberg, E., Jones, F.D., Horton, H.L., Ryffel, H.H., and McCauley, C.J. (2016): Machinery's Handbook: A Reference Book for the Mechanical Engineer, Designer, Manufacturing Engineer, Draftsman, Toolmaker and Machinist. 30th edition, Industrial Press, New York.
24. Oberlender, G. D. (2014): Project management for engineering and construction. 3rd ed. New York, N.Y.: McGraw-Hill Education LLC.
25. Rachuri, S., Han, Y.H., Feng, S.C., Wang, F., Sriram, R.D., Lyons, K.W. and Roy, U. (2003): Object-Oriented Representation of Electro-Mechanical Assemblies Using UML. Proceedings of the IEEE International Symposium on Assembly and Task Planning 2003: 228-234.
26. Ramaji, Issa J, S M Asce, Ali M Memari, and F Asce. (2016): Product Architecture Model for Multistory Modular Buildings. Journal of Construction Engineering and Management 142(10), 04016047.
27. Rogers, E. (2018): Phases of Architectural Design. https://www.wagstaffrogerarch.com/blog/phases-architectural-design/. Retrieved on April 11, 2019.
28. Russell, S. J, and Norvig, P. (1995). Artificial intelligence : a modern approach. Englewood Cliffs, N.J.: Prentice Hall.
29. Steward, D. (1981): The Design Structure System: A Method for Managing the Design of Complex Systems. IEEE Transactions on Engineering Management (3): 71-74.
30. Suh, N. P. (1995): Axiomatic Design of Mechanical Systems. Journal of Mechanical Design 117(June), 2-10.
31. Suh, N. P. (1997): Design Systems. Annals of the CIRP 46(1): 75-80.
32. Suh, N. P. (1998): Axiomatic Design Theory for Systems. Research in Engineering Design - Theory, Applications, and Concurrent Engineering 10(4), 189-209.
33. Tang, D. B., Zhang, G.J, and Dai, S. (2008):Integration of Axiomatic Design and Design Structure Matrix for Product Design. Advanced Materials Research 44-46, 421-428.
34. Tang, D. B., Zhang, G., and Dai, S. (2009): Design as Integration of Axiomatic Design and Design Structure Matrix. Robotics and Computer-Integrated Manufacturing 25(3), 610-619.
35. Tolman, F. P. (1999): Product Modeling Standards for the Building and Construction Industry: Past, Present and Future. Automation in construction 8(3): 227235.
36. Torry-smith, J. M., Achiche, S., Mortensen, N. M., Qamar, A., Wikander, J., and During, C. (2011): Mechatronic Design: Still a Considerable Challenge. Proceedings of the ASME 2011 International Design Engineering Technical Conferences - Computers and Information Engineering Conference, 33-44.
37. Uppal, K, B. (2001): Easy Factored Estimating and Process Cost Engineering. AACE International Transactions, 1-6.
38. Valdes, F., Gentry, R., Eastman, C., and Forrest, F. (2016): Applying Systems Modeling Approaches to Building Construction. 33rd International Symposium on Automation and Robotics in Construction (ISARC 2016).

39. Zein, R. (2011): Lessons Learned in Engineering Development for Major Projects, presented at the 55th Annual Meeting of AACE International, Anaheim, California.

Appendix A: List of Abbreviations

AD	Axiomatic design
CAD	Computer-aided design
CR	Customer requirement
DM	Design matrix
DP	Design parameter
DSM	Design structure matrix
FEED	Front-end engineering design
FR	Functional requirement
IFM	Integrated function modeling
MBSE	Model-based systems engineering
PV	Process variable
SCADA	Supervisory control and data acquisition
SISO	Single-input and single-output
SYSML	Systems modeling language
UML	Unified modeling language

Appendix B: Figures

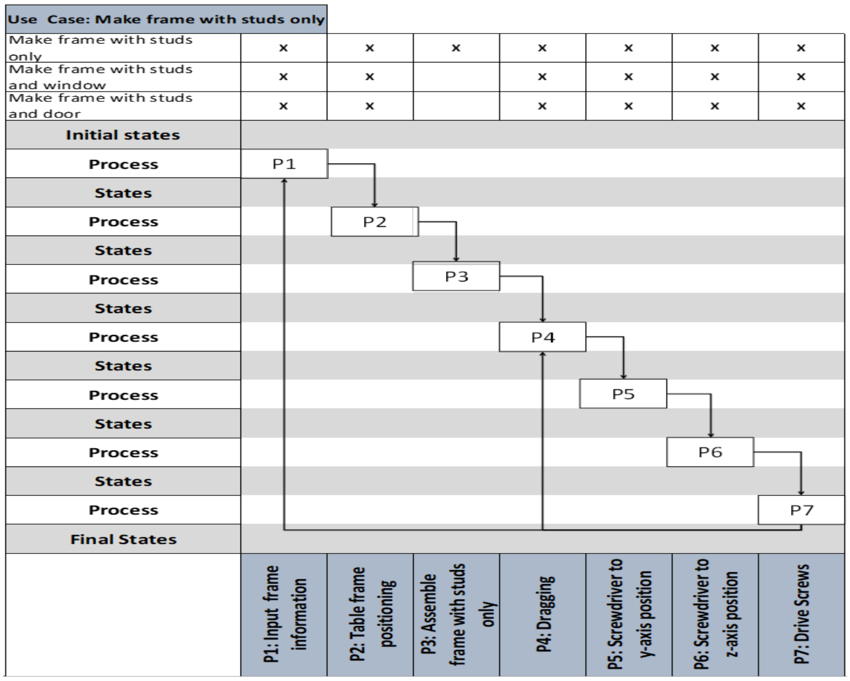

Fig. 10 Use case and process views of the automated steel wall framing assembly IFM framework.

		P1: Input frame information	P2: Table frame positioning	P3: Assemble frame with studs only	P4: Dragging	P5: Screwdriver to y-axis position	P6: Screwdriver to z-axis position	P7: Drive Screws
Automated steel wall frame asssembly	Actors							
Machine for 3 steel frame types	Actors							
User Interface	Actors	X						
Control Interface	Actors	X	X					
Recipe Data	Actors	X	X					
Right table movement	Actors		X					
Right table mechanism	Actors		X					
Right table mechanism control	Actors		X					
Manual assembly	Actors		X	X				
Squaring system	Actors			X	X			X
Holding mechanism	Actors			X	X			X
Holding control	Actors			X	X			X
Right angle positioning system	Actors			X	X			X
Right angle positioning control	Actors			X	X			X
Dragging system	Actors			X	X			X
Dragging system control	Actors			X	X			X
Y-axis screw driver positioning system	Actors					X		
Y-axis screw driver positioning control	Actors					X		
Z-axis screw driver positioning system	Actors						X	
Z-axis screw driver positioning control	Actors						X	
Fastening system	Actors						X	X
Fastening system control	Actors						X	X
Z-axis thrust position	Actors						X	X
Z-axis thrust position control	Actors						X	X
Operator	Actors	X	X	X	X		X	X
Electricity	Operands	X	X	X	X		X	X
Control System Commands	Operands	X	X	X	X		X	X
Steel Channel	Operands		O	O	O			O
80/20 Profile	Operands							
Screws	Operands							O

Fig. 11 Actor view of the automated steel wall framing assembly IFM framework.

	Screws	80/20 Profile	Steel Channel	Control System Commands	Electricity	Operator	Z-axis thrust postion control	Z-axis thrust postion	Fastening system control	Fastening system	Z-axis screw driver positioning ctl	Z-axis screw driver positioning sys	Y-axis screw driver positioning ctl	Y-axis screw driver positioning system	Dragging system ctl	Dragging system	Right angle positioning ctl	Right angle positioning sys	Holding control	Holding mechanism	Squaring system	Manual assembly	Right table mech control	Right table mechanism	Right table movement	Recipe Data	Control Interface	User Interface	Machine for 3 steel frame types	Automated steel wall frame asm	
Automated steel wall frame asm		x	x	x	x																									x	Actors
Machine for 3 steel frame types		x	x	x	x																								x		Actors
User Interface					x	x																				x	x	x			Actors
Control Interface				x	x																					x	x				Actors
Recipe Data				x																						x					Actors
Right table movement				x	x																				x						Actors
Right table mechanism				x	x																		x	x							Actors
Right table mech control				x	x																		x	x							Actors
Manual assembly			x			x																x									Actors
Squaring system			x	x	x																x										Actors
Holding mechanism			x	x	x														x	x											Actors
Holding control			x	x	x														x	x											Actors
Right angle positioning sys			x	x	x												x	x													Actors
Right angle positioning ctl			x	x	x												x	x													Actors
Dragging system			x	x	x										x	x															Actors
Dragging system control			x	x	x										x	x															Actors
Y-axis scdr positioning sys			x	x	x								x	x																	Actors
Y-axis scdr positioning ctl			x	x	x								x	x																	Actors
Z-axis scdr positioning sys			x	x	x						x	x																			Actors
Z-axis scrdr positioning ctl			x	x	x						x	x																			Actors
Fastening system	x		x	x	x			x	x	x																					Actors
Fastening system control	x		x	x	x				x	x																					Actors
Z-axis thrust postion	x		x	x	x		x	x																							Actors
Z-axis thrust position control	x		x		x		x	x																							Actors
Operator						x																									Actors
Electricity			x		x																										Operands
Control System Commands			x	x																											Operands
Steel Channel			x																												Operands
80/20 Profile		x																													Operands
Screws	x																														Operands
	Operands					Actors																									

Fig. 12 Interaction view of the automated steel wall framing assembly IFM framework.

Automated steel wall frame assembly	Actors															active
Machine for three steel frame types																active
User Interface			Supporting P7		Supporting P6		Supporting P5		Supporting P4		Supporting P3		Supporting P2		P1	active
Control Interface			Supporting P7		Supporting P6		Supporting P5		Supporting P4		Supporting P3		Supporting P2		P1	active
Recipe Data			Supporting P7		Supporting P6		Supporting P5		Supporting P4		Supporting P3		Supporting P2	accessed	P1	upload
Right table movement												inactive	P2	active		inactive
Right table mechanism												inactive	P2	active		inactive
Right table mechanism control												inactive	P2	active		inactive
Manual assembly										standby	P3	active				active
Squaring system									P4	active						inactive
Holding mechanism									P4	active						inactive
Holding control									P4	active						inactive
Right angle positioning system									P4	active						inactive
Right angle positioning control									P4	active						inactive
Dragging system			Supporting P7		Supporting P6		Supporting P5		P4	active						inactive
Dragging system control			Supporting P7		Supporting P6		Supporting P5		P4	active						inactive
Y-axis screw driver positioning system						inactive	P5	active								inactive
Y-axis screw driver positioning control						inactive	P5	active								inactive
Z-axis screw driver positioning system			Supporting P7		P6	active										inactive
Z-axis screw driver positioning control			Supporting P7		P6	active										inactive
Fastening system				active												inactive
Fastening system control			P7	active												inactive
Z-axis thrust postion			P7	active												inactive
Z-axis thrust postion control			P7	active												active
Operator	Operands		Supporting P7		Supporting P6		Supporting P5		Supporting P4		Supporting P3		Supporting P2		Supporting P1	active
Electricity			Supporting P7		Supporting P6		Supporting P5		Supporting P4		Supporting P3		Supporting P2		Supporting P1	active
Control System			Supporting P7		Supporting P6		Supporting P5		Supporting P4		Supporting P3		Supporting P2		Supporting P1	active
Steel Channel		fastened frame								soft connected						channels
80/20 Profile			Supporting P7		Supporting P6		Supporting P5		Supporting P4		Supporting P3		Supporting P2		Supporting P1	installed
Screws		driven	P7	drive screw												inactive

Fig. 13 State view of the automated steel wall framing assembly IFM framework.

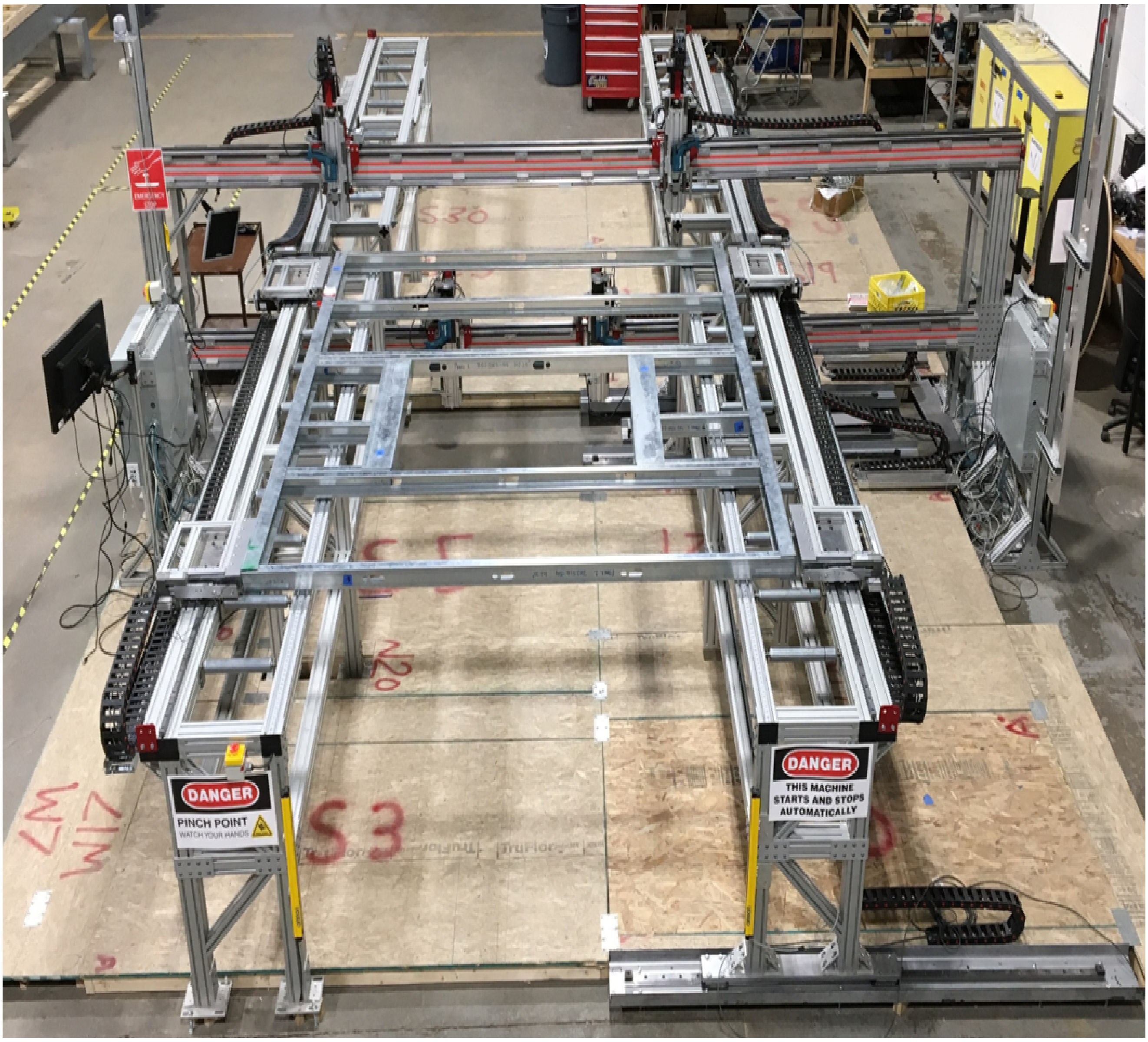

Fig. 14 Realization of the automated steel wall framing machine discussed in sections 3 and 4.